

License Generator 開発マニュアル

2009年8月17日

株式会社 アイ・ティー・ワン

新規事業開発本部 営業部

はじめに

License Generator をお買い上げいただきましてありがとうございます。

本書は、開発されたパッケージソフトウェアにセキュリティおよびライセンス機能を実装する手順について以下の流れで説明いたします。

- | | | | |
|-----|---------------------------------|-----|-------|
| (1) | License Generator を使用するために必要な情報 | ・・・ | p. 3 |
| (2) | ライセンス認証ファイルの発行 | ・・・ | p. 4 |
| (3) | パッケージソフトのソースコードへの記述方法 | ・・・ | p. 5 |
| (4) | パッケージソフトのインストーラ作成時の注意事項 | ・・・ | p. 13 |
| (5) | ライセンス認証に関する仕様 | ・・・ | p. 14 |
| (6) | インターネット認証 | ・・・ | p. 15 |
| (7) | 関数仕様 | ・・・ | p. 17 |

License Generator の仕様は予告なく変更する場合があります。

1. License Generator を使用するために必要な情報

以下の情報を確認してください。

No	情報名	説明	○
1	パッケージソフトウェアの製品名	プロテクション対象の製品名	
2	ライセンスデータベースの格納フォルダ名	指定されたフォルダに発行したライセンス情報が一元管理されます。	
3	発行したライセンスの格納フォルダ名	指定されたフォルダにライセンスファイル「licence.ini」が作成されます。	
4	ライセンス認証サーバの IP アドレス	インターネットでライセンス認証する場合に必要です。	
5	ライセンス認証サーバの通信ポート番号	デフォルトでは 44400 番を使用します。	
6	プロテクション情報	License Generator のライセンス発行画面で設定します。	

これらの必要な情報をもとに、パッケージソフトウェアにプロテクションをかけます。

本書では、以下の**サンプル製品**をプロテクトすることを想定し具体的な手順で説明します。

No	情報名	情報詳細	○
1	パッケージソフトウェアの製品名	HELLOWORLD	
2	ライセンスデータベースの格納フォルダ名	C:¥ライセンス管理	
3	発行したライセンスの格納フォルダ名	C:¥ライセンス発行	
4	ライセンス認証サーバの IP アドレス	172.23.10.200	
5	ライセンス認証サーバの通信ポート番号	44400	
6	プロテクション情報	ライセンス名は "X X X X 用評価版" 試用期間は 6 0 日。 インターネット認証を行う。 発行ライセンスは 1 6 とする。	

※License Generator のインストールフォルダ : "C:¥Program Files¥License Generator"

Activation Server のインストールフォルダ : "C:¥Program Files¥Activation Server"

2. ライセンス認証ファイルの発行

- (1) License Generator のインストールフォルダにある"ENVIRON.DEF"の内容を書き換えます。

[DB_DIRECTORY] C:¥ライセンス管理	ライセンスデータベースを格納するフォルダ名を記述します。	ライセンスジェネレータでは、ライセンス管理 DB はローカルディスクまたはネットワークドライブに設定してください。
[SERVER_NAME] 172.23.10.200	ライセンス認証サーバのグローバルIPまたはドメイン名を記述します。	
[SERVER_PORT] 44400	ライセンス認証サーバの通信ポート番号を記述します。	

- (2) License Generator のインストールフォルダにある"ライセンスジェネレータ起動.BAT"の内容を書き換えます。

```
@ECHO OFF
START GENLCS.exe "HELLOWORLD" "C:¥ライセンス発行"
```

プログラムの引数に「製品名」と「ライセンス発行先フォルダ」を指定します。

- (3) License Generator を起動しライセンスファイルを作成します。

スタート ⇒ プログラム ⇒ License Generator ⇒ ライセンスジェネレータ起動.BAT

※ユーザ名とパスワードは、root / root です。

The screenshot shows the License Generator application window. The product name is set to 'HELLOWORLD'. The license name field contains 'XXXX用評価版'. The trial period is set to 0060 days. The license server IP is 127.23.10.200 and the port is 44400. The number of licenses to generate is 16. The interface includes fields for license keys, a search function, and a 'Generate License File' button.

① ライセンス名を指定したら、まず登録ボタンをクリックして認証番号を発行してください。

② 赤い丸で指定したライセンス情報を設定してください。

③ 設定が終了したら、ライセンス認証ファイル作成をします。

顧客情報[XXXX用評価版]の登録が完了しました。

3. パッケージソフトのソースコードへの記述方法

パッケージソフト「HELLOWORLD」をVC++（MFC）ダイアログベースで作成したとします。

プログラム名を" UNEMPCAL"とした場合には、" UNEMPCAL.cpp "と" UNEMPCALDlg.cpp "のソースコードが作成されます。

プロテクションの検査はダイアログを表示する前に行いますので、" UNEMPCAL.cpp "に以下の記述をしてください。

```
//-----  
// プロトタイプ宣言  
//-----  
int  CipherNumber (void); // ワンタイムパスワードを取得  
int  Confirmation (char *); // ライセンスの認証  
int  LicenseEntry (void); // ライセンス番号の入力要求  
char *GetOption (void); // オプションの取得  
char *GetInformation (void); // 付加情報の取得  
  
BOOL CUNEMPCALApp::InitInstance() {  
    int Rc, Ans;  
    CWinApp::InitInstance();  
    //=====  
    // ライセンスの検査 開始  
    //=====  
    Ans = CipherNumber(); // ワンタイムパスワード取得  
    Rc = Confirmation("HELLOWORLD"); // ライセンス認証結果を取得  
    if (Ans != Rc) { // 認証結果がワンタイムパスワードと一致しない：認証エラー  
        switch (Rc) {  
            case -1:  
                MessageBox(GetFocus(), "ライセンス認証ファイルがありません", "認証エラー", MB_OK);  
                return FALSE;  
            case -2:  
                MessageBox(GetFocus(), "ライセンス認証ファイルにアクセス権がありません", "認証エラー", MB_OK);  
                return FALSE;  
            case -4:  
                MessageBox(GetFocus(), "ライセンス認証ファイルが改竄されています", "認証エラー", MB_OK);  
                return FALSE;  
            case -8:  
                MessageBox(GetFocus(), "ライセンス認証ファイルは当製品のものではありません", "認証エラー", MB_OK);  
                return FALSE;  
            case -16:  
                MessageBox(GetFocus(), "この端末では製品の実行が許可されていません", "認証エラー", MB_OK);  
                return FALSE;  
            case -32:  
                MessageBox(GetFocus(), "他の端末で認証されたライセンスは使用できません", "認証エラー", MB_OK);  
                return FALSE;  
            case -64: // 試用版は複数回インストールすることはできません（製品版の認証を要求する）  
                if (LicenseEntry()) { return FALSE; } else { break; }  
            case -128: // 試用期間が終了しました（製品版の認証を要求する）  
                if (LicenseEntry()) { return FALSE; } else { break; }  
            default :  
                return FALSE;  
        }  
    }  
}
```

```

BOOL CUNEMPCALDlg::OnInitDialog() {
    CDialog::OnInitDialog();
    SetIcon(m_hIcon, TRUE);
    SetIcon(m_hIcon, FALSE);

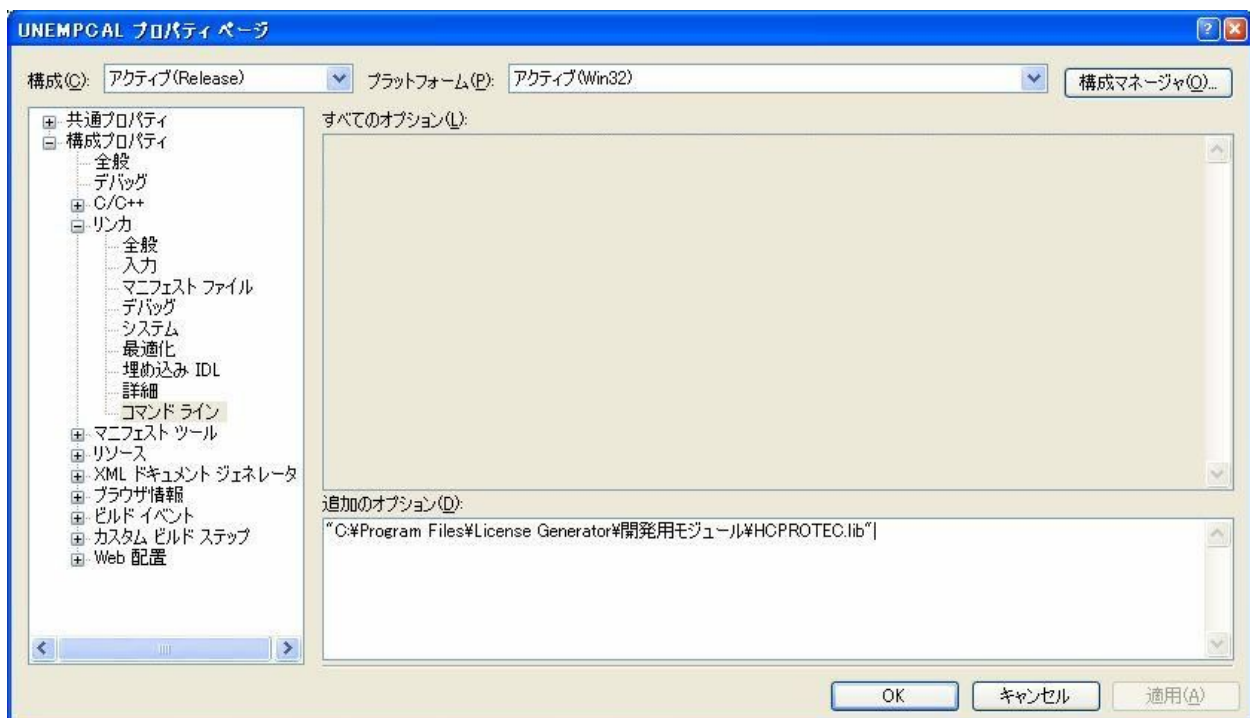
    char *GetOption(void); // オプションの取得
    char *GetInformation(void); // 付加情報の取得
    //-----
    // オプション情報の取得
    //-----
    m_Option.SetWindowTextA(GetOption());
    //-----
    // 任意情報の取得
    //-----
    m_Infomation.SetWindowTextA(GetInformation());
    return TRUE;
}

```

ソースコードの記述が終わったならば、以下のモジュールをパッケージ製品のリンク時に指定してください。

●C:¥Program Files¥License Generator¥開発用モジュール

HCPROTEC.lib



VB で実装する場合は次のコードを参考にして記述してください。

関数定義場所、認証エラー時のハンドリングは適宜アプリケーションのつくりに合わせてください。

```
Imports System.Runtime.InteropServices
Public Class Class1
#Region "DLL"
    <DllImport("HCPROTEC.dll")> _
    Private Shared Function CipherNumber() As Integer
    End Function
    <DllImport("HCPROTEC.dll")> _
    Private Shared Function Confirmation(ByVal proname As String) As Integer
    End Function
    <DllImport("HCPROTEC.dll")> _
    Private Shared Function LicenseEntry() As Boolean
    End Function
#End Region
    Public Shared Sub main()
        Dim Ans As Integer = CipherNumber()
        Dim Rc As Integer = Confirmation("HELLOWORLD")
        If Ans <> Rc Then
            Select Case Rc
                Case -1
                    MessageBox.Show("ライセンス認証ファイルがありません", "認証エラー")
                    Return
                Case -2
                    MessageBox.Show("ライセンス認証ファイルにアクセス権がありません", "認証エラー")
                    Return
                Case -4
                    MessageBox.Show("ライセンス認証ファイルが改竄されています", "認証エラー")
                    Return
                Case -8
                    MessageBox.Show("ライセンス認証ファイルは当製品のものはありません", "認証エラー")
                    Return
                Case -16
                    MessageBox.Show("この端末では製品の実行が許可されていません", "認証エラー")
                    Return
                Case -32
                    MessageBox.Show("他の端末で認証されたライセンスは使用できません", "認証エラー")
                    Return
                Case -64 ' 試用版は複数回インストールすることはできません (製品版の認証を要求する)
                    If LicenseEntry() Then
                        Return
                    Else
                        Exit Select
                    End If
                Case -128 ' 試用期間が終了しました (製品版の認証を要求する)
                    If LicenseEntry() Then
                        Return
                    Else
                        Exit Select
                    End If
                Case Else
                    Return
            End Select
        End If
    End Sub
End Class
```

```
Imports System.Runtime.InteropServices
Public Class Form1
    Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button1.Click
        MessageBox.Show("Hello World!", "License Generator")
    End Sub

    Private Sub Button2_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button2.Click
        Me.Close()
    End Sub

    Private Sub Form1_Load(ByVal sender As Object, ByVal e As System.EventArgs) Handles Me.Load
        textBox3.Text = HCPROTEC.HCOPTINF.GetOption()
        TextBox4.Text = HCPROTEC.HCOPTINF.GetInformation()
    End Sub
End Class
```

Delphi で実装する場合は次のコードを参考にして記述してください。

関数定義場所、認証エラー時のハンドリングは適宜アプリケーションのつくりに合わせてください。

```
unit Sample;
<中略>
//-----
// プロトタイプ宣言
//-----
{ワンタイムパスワードを取得}
function CipherNumber() : Integer; stdcall;
external 'HCPROTEC.dll' name 'CipherNumber';
{ライセンスの認証}
function Confirmation(cpProductName: PChar) : Integer; stdcall;
external 'HCPROTEC.dll' name 'Confirmation';
{ライセンス番号の入力要求}
function LicenseEntry() : Integer; stdcall;
external 'HCPROTEC.dll' name 'LicenseEntry';
{オプションの取得}
function GetOption() : PChar; stdcall;
external 'HCPROTEC.dll' name 'GetOption';
{付加情報の取得}
function GetInformation() : PChar; stdcall;
external 'HCPROTEC.dll' name 'GetInformation';
<中略>
//-----
// ライセンスの認証
//-----
function Authenticate(): Boolean;
var Rc : Integer;
    Ans : Integer;
    rnt : Integer;
    ProductName : PChar;
begin
    ProductName := 'HELLOWORLD';      { 製品名 }
    Ans := CipherNumber();            { ワンタイムパスワード取得 }
    Rc := Confirmation(ProductName); { ライセンス認証結果を取得 }
    { 最適化を適用する場合は以下のコードを記述する }
    //IntToStr(Rc);
    //IntToStr(Ans);
    if(Ans <> Rc) then { 認証結果がワンタイムパスワードと一致しない: 認証エラー }
    begin
        { 最適化を適用する場合は以下のコードを記述する }
        //IntToStr(Rc);
        //IntToStr(Ans);
        case Rc of
            -1:
                begin
                    ShowMessage('ライセンス認証ファイルがありません');
                    Result := False;
                end;
            -2:
                begin
                    ShowMessage('ライセンス認証ファイルにアクセス権がありません');
                    Result := False;
                end;
        end;
    end;
end;
```

```

-4:
begin
  ShowMessage('ライセンス認証ファイルが改竄されています');
  Result := False;
end;
-8:
begin
  ShowMessage('ライセンス認証ファイルは当製品のものではありません');
  Result := False;
end;
-16:
begin
  ShowMessage('この端末では製品の実行が許可されていません');
  Result := False;
end;
-32:
begin
  ShowMessage('他の端末で認証されたライセンスは使用できません');
  Result := False;
end;
-64: { 試用版は複数回インストールすることはできません (製品版の認証を要求する) }
begin
  if LicenseEntry() <> 0 then
  begin
    Result := False;
  end
  else
  begin
    Result := True;
  end
end;
-128: { 試用期間が終了しました (製品版の認証を要求する) }
begin
  if LicenseEntry() <> 0 then
  begin
    Result := False;
  end
  else
  begin
    Result := True;
  end
end;

else
begin
  Result := False;
end;
end;
end;
end
else
begin
  Result := True;
end;
end;
// オプションと任意情報を取得
Edit1.Text := GetOption();
Edit2.Text := GetInformation();
{ 処理を続く...}

```

C#で実装する場合は次のコードを参考にして記述してください。

関数定義場所、認証エラー時のハンドリングは適宜アプリケーションのつくりに合わせてください。

```
using System;
using System.Collections.Generic;
using System.Windows.Forms;
using System.Runtime.InteropServices;
namespace HELLOWORLD {
    static class Program    {
        #region DLL
        [DllImport("HCPROTEC.dll")]
        private static extern int CipherNumber();
        [DllImport("HCPROTEC.dll")]
        private static extern int Confirmation(string proname);
        [DllImport("HCPROTEC.dll")]
        private static extern bool LicenseEntry();
        #endregion
        /// <summary>
        /// アプリケーションのメイン エントリ ポイントです。
        /// </summary>
        [STAThread]
        static void Main()    {
            int Ans = CipherNumber();           // ワンタイムパスワード取得
            int Rc  = Confirmation("HELLOWORLD"); // 製品名を指定してライセンス認証結果取得
            if (Ans != Rc)
            { // 認証結果がワンタイムパスワードと一致しない：認証エラー
                switch (Rc) {
                    case -1: MessageBox.Show("ライセンス認証ファイルがありません", "認証エラー");
                        return;
                    case -2: MessageBox.Show("ライセンス認証ファイルにアクセス権がありません", "認証エラー");
                        return;
                    case -4: MessageBox.Show("ライセンス認証ファイルが改竄されています", "認証エラー");
                        return;
                    case -8: MessageBox.Show("ライセンス認証ファイルは当製品のものではありません", "認証エラー");
                        return;
                    case -16: MessageBox.Show("この端末では製品の実行が許可されていません", "認証エラー");
                        return;
                    case -32: MessageBox.Show("他の端末で認証されたライセンスは使用できません", "認証エラー");
                        return;
                    case -64: // 試用版は複数回インストールすることはできません (製品版の認証を要求する)
                        if (LicenseEntry()) { return; } else { break; }
                    case -128: // 試用期間が終了しました (製品版の認証を要求する)
                        if (LicenseEntry()) { return; } else { break; }
                    default:
                        return;
                }
            }
            //Application.EnableVisualStyles();
            Application.SetCompatibleTextRenderingDefault(false);
            Application.Run(new Form1());
        }
    }
}
```

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Text;
using System.Windows.Forms;
using System.Runtime.InteropServices;

namespace HELLOWORLD {
    public partial class Form1 : Form {
        public Form1() {
            InitializeComponent();
            this.textBox3.Text = HCPROTEC.HCOPTINF.GetOption();
            this.textBox4.Text = HCPROTEC.HCOPTINF.GetInformation();
        }
        private void button1_Click(object sender, EventArgs e)
        {
            MessageBox.Show("    Hello.World!", "License Generator");
        }
        private void button2_Click(object sender, EventArgs e)
        {
            this.Close();
        }
    }
}
```

4. パッケージソフトのインストーラ作成時の注意事項

パッケージソフトのインストーラ作成時には、インストールフォルダ配下にある以下のファイルと作成したライセンス認証ファイルを同梱してください。

- C:¥Program Files¥License Generator¥配布モジュール

HCPROTEC.dll (必須)

HCOPTINF.dll (C#,VBのみ必須。C++,DELPHIでは不要)

SETLCS.exe (必須)

- C:¥ライセンス発行¥HELLOWORLD¥[ライセンス番号]

LICENSE.ini (必須)

※上記のファイルは、製品の実行体と同じフォルダに配置するようにパッケージしてください。

※ライセンス認証ファイル (LICENSE.ini) はライセンス番号のサブフォルダに生成されます。インストーラ作成時には該当するライセンスのライセンス認証ファイルを組み込むように操作してください。

パッケージソフトのインストーラ作成時には、ファイルシステムのサブフォルダを含む全てのフォルダのプロパティ : AlwaysCreate を False にしてください。



5. ライセンス認証に関する仕様

NO	区分	説明
1	試用期限	試用期限を越えて評価版は使用できない。
2	試用期限	試用期限を経過した後にコンピュータ日付を戻しても使用できない。
3	試用期限	試用期限を経過した後にレジストリを削除して再インストールしても使用できない。
4	試用日数	初回試用日から指定日数を越えて評価版は使用できない。
5	試用日数	試用日数を経過して後にコンピュータ日付を戻しても使用できない。
6	試用日数	試用日数を経過した後にレジストリを削除して再インストールしても使用できない。
7	ライセンス認証ファイル	ライセンス認証ファイルがなければ使用できない。
8	ライセンス認証ファイル	ライセンス認証ファイルに読み込み権がなければ使用できない。
9	ライセンス認証ファイル	改竄されたライセンス認証ファイルでは使用できない。
10	必須モジュール	プロテクトDLLがなければ使用できない。
11	必須モジュール	ライセンス認証プログラムがなければ使用できない。
12	ライセンス認証済	レジストリが削除されても使用できる。
13	ライセンス認証済	認証済のライセンスファイルは、他のPCで使用できない。
14	自製品でないライセンス	他製品のライセンス認証ファイルは使用できない。
15	特定端末	実行可能端末が指定されている場合は、他の端末では使用できない。
16	インターネット認証	インターネット認証が指定された場合は、インターネット認証する。
17	インターネット認証	ライセンス認証サーバが起動されていない場合は認証できない。
18	インターネット認証	指定ライセンス数を越えてインターネット認証できない。
19	インターネット認証	指定ライセンス数を書き換えることで、認証ライセンス数を増やすことができる。
20	インターネット認証	ライセンス認証サーバが指定時間応答しないとタイムアウトになる。
21	インターネット認証	指定ポートに、他の製品からアクセスすると、エラーログが採取される。
22	ジェネレータ	発行したライセンスは、部分名または部分番号で検索できる。
22	ジェネレータ	他の製品のデータベースは読み込めない
23	認証サーバ	WINDOWS が起動されれば、認証サーバは起動する。(モニターとは無関係)
24	動作環境	動作環境は、ENVIRON.DEF (環境定義ファイル) で定義される。

※レジストリ値：HKEY_LOCAL_MACHINE¥SOFTWARE¥製品名に、ライセンスの有効期限および、最終実行日が記録されます。レジストリ値とライセンス認証ファイルの値は漸化式で検査しますので、レジストリ値を意図的に操作すると有効期限が終了します。

6. インターネット認証

(1) Activation Server のインストールフォルダにある"ENVIRON.DEF"の内容を書き換えます。

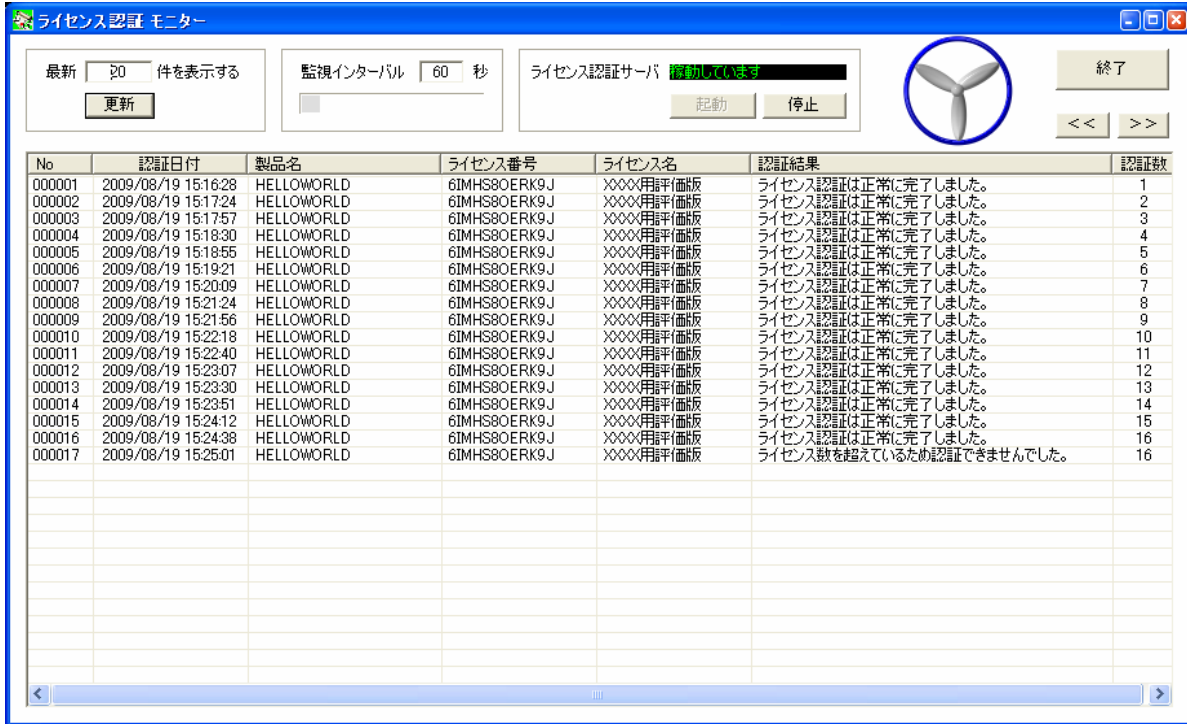
[DB_DIRECTORY] C:\ライセンス管理	ライセンスデータベースを格納するフォルダ名を記述します。	アクティベーションサーバでは、ライセンス管理 DB はローカルディスクに設定してください。
[SERVER_PORT] 44400	ライセンス認証サーバの通信ポート番号を記述します。	

スタート ⇒ プログラム ⇒ License Generator ⇒ Activation Server

初回起動時は登録をクリックしてください。自動起動タスクを登録します。
起動/停止をクリックすると、認証サーバが起動/停止します。
ライセンス認証サーバが起動状態になっていれば、ライセンス認証モニターを終了しても、認証サーバはバックグラウンドで動作します。



製品のライセンス認証が行われると、ライセンス認証モニターに表示されます。



※Activation Server は、「ライセンス認証サーバで認証する」と指定した製品が、ライセンス認証を行うためのものです。

7. 関数仕様

機能名

実行毎にワンタイムパスワードを発行する

関数仕様

int CipherNumber (void);

復帰値

1～2147483648 までのいずれかの数値を返却します。

機能名

指定された製品のライセンス認証を行う

関数仕様

int Confirmation (char *製品名);

引数

製品名は **Shift_Jis** コードで関数に渡してください。

復帰値

ライセンス認証が正常ならば、**CipherNumber** 関数で取得した値と同じ値を返却します。

ライセンス認証できない場合は、以下の負の値を返却します。

- 1:ライセンス認証ファイルがありません。
- 2:ライセンス認証ファイルにアクセス権がありません。
- 4:ライセンス認証ファイルが改竄されています。
- 8:ライセンス認証ファイルは当製品のものではありません。
- 16:この端末では製品の実行が許可されていません。
- 32:他の端末で認証されたライセンスは使用できません。
- 64:試用版は複数回インストールすることはできません。
- 128: 試用期間が終了しました。

機能名

ライセンス番号の入力画面を表示しライセンス認証を行う

関数仕様

int LicenseEntry (void);

復帰値

正しいライセンス番号が入力された場合は、0を返却します。

正しいライセンス番号が入力されない場合は、1を返却します。

機能名

C++、Delphi 言語で License Generator で指定したオプション情報を取得する。

関数仕様

```
char *GetOption (void);
```

復帰値

8 個のオプション文字列が Shift_Jis コードの 0 と 1 で返却されます。
オプションが指定されなかった場合は、"00000000" が返却されます。
全てのオプションが指定された場合は、"11111111" が返却されます。

機能名

C#、VB 言語で License Generator で指定したオプション情報を取得する。

関数仕様

```
Text HCPROTEC.HCOPTINF.GetOption();
```

復帰値

8 個のオプション文字列が Shift_Jis コードの 0 と 1 で返却されます。
オプションが指定されなかった場合は、"00000000" が返却されます。
全てのオプションが指定された場合は、"11111111" が返却されます。

機能名

C++、Delphi 言語で License Generator で指定した任意の情報を取得する。

関数仕様

```
char *GetInformation (void);
```

復帰値

指定した任意の文字列が Shift_Jis コードで返却されます。
文字列が指定されていない場合は、ナル文字が返却されます。

機能名

C#、VB 言語で License Generator で指定した任意の情報を取得する。

関数仕様

```
Text HCPROTEC.HCOPTINF.GetInformation ();
```

復帰値

指定した任意の文字列が Shift_Jis コードで返却されます。
文字列が指定されていない場合は、ナル文字が返却されます。

【お客さまが作成した評価版の製品を一般ユーザ権限で評価させるには】

License Generator は、製品の評価段階で、ライセンスファイル「LICENSE.ini」とレジストリ「HKEY_LOCAL_MACHINE¥SOFTWARE¥製品名」を参照・更新いたします。

そのため、評価段階では administrator グループの権限を持つユーザでのみ実行可能です。

評価版の製品を、一般ユーザで評価・実行させたい場合は、CD-ROM に同封されている「UNSETREL.exe」をインストーラの終了時の起動プログラムとして実行してください。

→ UNSETREL.exe “製品名”

※ Visual Studio のパッケージ機能では、「カスタム動作→確定」で指定できます。

UNSETREL.exe は、ライセンスファイルとレジストリのアクセス権限を一般ユーザに与えます。

【重要】

製品版の License Generator は、ご購入頂いたお客様毎に個別管理いたします。

- ・ **暗号化の基底情報**はお客様毎に管理されます。したがって他社が使用する License Generator が発行するライセンス認証ファイル等、データの**互換性は一切発生いたしません**。
- ・ **提供関数名**もお客様毎に管理されます。（上記の関数名は評価版でのみ有効です）
提供関数名は、お客様のご使用になる開発言語の**サンプルプログラム**内に記述してあります。

License Generator お問い合わせ先

株式会社アイ・ティー・ワン 新規事業開発本部 営業部

e-it1newbiz@it-one.co.jp

株式会社アイ・ティー・ワン 〒160-0022 東京都新宿区新宿 2-8-8

<http://www.it-one.co.jp/index.html>

© 2009 IT-One Co., Ltd.